The Relational Database

What is a Database?

A database may be defined as any organized collection of information. For our purposes we will define a database as a set of related files that are created and managed by a Database Management System (DBMS).

What is a Relational Database?

The term relational database has been used to describe almost any database created since the 1980s. It can mean any database that stores data in tables that can establish relationships with other tables based on common information. More recently the definition of a relational database is any database which can be accessed using the SQL query language. Relational databases are what we will concern ourselves with in this paper.

Tables, Rows and Columns

It is difficult to discuss anything about databases without referring to tables, rows and columns. If you have a basic understanding of computer terminology you probably understand what we are talking about when we discuss a file, a record and a field. In database jargon you can think of a table as a file, a row as a record and a column as a field.

Distributed Databases

A Distributed Database is a database which is not contained on a single server but on several servers spread throughout an organization. For example, a customer table may be located on a server at corporate headquarters, while customer credit histories may be maintained at a credit office geographically removed from headquarters. Together the tables would form a complete database of customer information.

Distributed databases are complex and involve many more problems that the non-distributed variety. Data is often replicated from one table to another and the physical services and supporting software may vary from one location to another.

Locking mechanisms, updates, and changes to data are much more difficult to implement when information is distributed and replicated.

Despite the disadvantages of Distributed Databases, there are two primary reasons businesses continue to utilize distributed data. The first is performance. Generally the closer you can get the data to the user the greater the performance will be.

The second reason is corporate evolution. Few businesses spring up overnight. Client/Server technology is often used to link previously unconnected data, applications and systems. The ability to link distributed data saves the cost and expense of having to rebuild from the ground up.

Distribution of Data

Three common ways of distributing data are downloading, replication and fragmentation. Each has its own advantages and disadvantages and should not be thought of as mutually exclusive. It is often helpful to combine these methods of distribution. The best method or combination of methods is determined by the application and the way in which the data is used.

Downloading

Often the simplest and best method of data distribution is to reproduce it by downloading. An example would be an inventory database that is copied each night from a centralized computer location to branch offices. The inventory would then be available for local manipulation the following day.

Downloading is best used in situations where data does not change very often such as price lists, employee records and payroll services. An application which uses downloading should not be populated with information that is frequently updated or that may be considered mission critical on a daily basis. Executive Decision Support is a good example of an application in which downloading might be used.

Decision support requires “snapshots” of information in order to best evaluate business trends. If the Decision Support System were constantly updated every few seconds the users of the system would not be able to draw insightful conclusions.

Replication

Replication is the ability to duplicate key portions of a database in various locations, making sure that all copies of the data are simultaneously updated. Replication ensures that if a user updates a local copy of a table, the DBMS automatically and immediately updates all other copies regardless of their location. Replication is best used where the users must have the most current information.

Fragmentation

Fragmentation splits up portions of a table among several servers. Tables may be split horizontally or vertically.

Horizontal Fragmentation splits the table into two or more groups of rows and stores the rows on separate servers. An example might be a customer database in which customer names in the United States are kept on a server in New York and customer names from Europe are kept on another server located in London, England.

Vertical Fragmentation splits a table into two or more groups or columns. An employee table may be vertically split with an employee payroll server containing information needed to authorize payroll checks, while the remainder of the employee information resides on a separate server at another location.

Whichever way a table is fragmented, the DBMS should be able to transparently reassemble the table for the end user.

Data Integrity

One of the primary concerns for any database administrator (or anyone else using the database) is data integrity. Data integrity simply refers to the validity and ability to trust the data. Without integrity the data is meaningless. Some of the ways in which data integrity is maintained is through the use of constants and nulls.

Constants

Constants establish values for specific fields. The Database Management System (DBMS) automatically checks each field to determine if values fall outside of acceptable parameters. If the values are invalid the information is rejected. Checking for proper values is also called Validity Checking. Examples of constants are:

- Numeric quantities which may only be positive numbers.
- A state or country code which may only contain alpha characters.
- Social security or employee numbers which consist of nine numbers.

Nulls

A null field is simply a field which has no value or the value is unknown. Do not confuse this with a field which has a value of zero. For example, if we were to calculate the average score in a football game and some of the games resulted in a score of zero, those scores would have to be factored into the overall average. If the field contains a null or unknown value, it would not be included in the calculation.

Some fields may be set up to allow null values, particularly non-critical fields, whereas others would require a non-null value, as in the name field of a personnel database.

Referential Integrity

Referential integrity checks the relationships of data contained in one table with data referenced in another table. For example, if you had one table containing customer information and the other containing invoice information, the two tables might be joined by a customer number field.

Referential integrity answers questions which may come up in the following situations;

- If you delete the customer record, are the corresponding invoice records also deleted?
- What happens if you add an invoice with a customer number that does not exist in the customer table?
- If the customer number is changed are the invoice records updated?
- If the customers invoices are deleted is the customer deleted in the customer number table?

Declarative Referential Integrity and Triggers

Two ways of enforcing referential integrity is with declarative referential integrity and triggers. Declarative referential integrity builds constraints into the system when the database is initially designed. An example of a referential constraint might be that no invoice could be created without a customer number.

A trigger is a small program that is run whenever data is altered. The trigger program checks to make sure referential integrity has not been violated.

Declarative referential integrity can be though of as being proactive whereas triggers may be thought of as reactive. Given the choice, declarative referential integrity is the preferred methodology.

Business Rules

Business rules are rules which are unique to a particular organization and may reflect field validation and integrity. Examples of possible business rules are:

- Christmas is a paid holiday.
- Orders are accepted with some credit cards but not others.
- Each employee has four levels of wage rates.
- Orders are only accepted in US dollars.

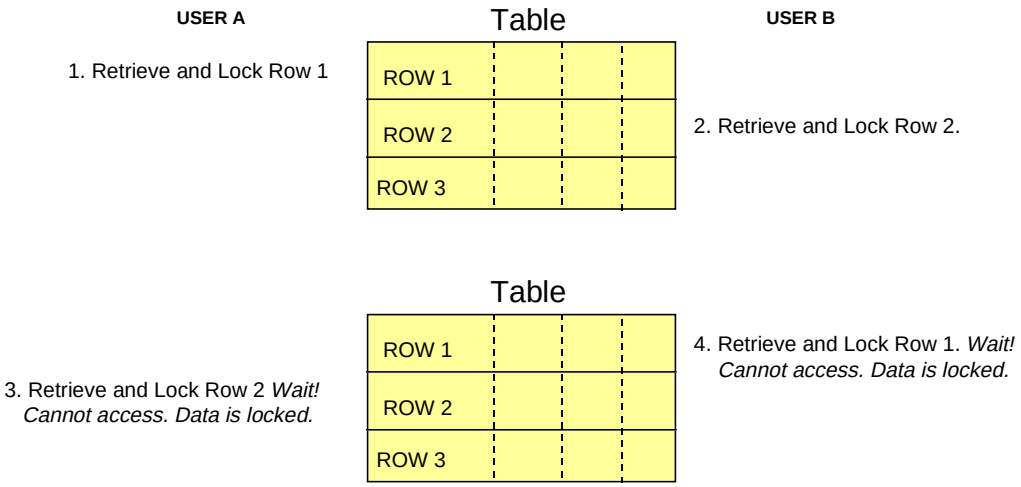
Concurrency and Locking

Concurrency is one of the most challenging aspects of developing client/server database applications. Concurrency allows multiple users to access a database at the same time. SQL does this by using "locks." Locking ensures that users do not interfere with one another. As one user accesses and manipulates the data other users are locked out and prevented from accessing the information until the lock is released.

Most Database Management Systems (DBMS) lock an entire page of information. A problem with this is that the page may contain data which is needed by others accessing the database. The information on the page is essentially unavailable until the lock is released, thereby slowing access and updates to the database.

There are two different kinds of locks: shared and exclusive. A *shared* lock is a read-only lock that allows users to read the data but not update it. An *exclusive* lock prevents users from reading or updating the data until the lock is released.

A potential deadlock situation may occur when two users are waiting for resources held by the other. In the following situation User A begins by retrieving and locking Row1 and User B retrieves and locks Row 2. Next User 2 attempts to read Row 2, but must wait until User B releases the lock on Row 2. Next User 2 attempts to read Row 1 which is locked by User A. At this point a deadlock occurs. Neither User A nor User B can access the data they need until one releases their lock. At this point the DBMS should intervene and cancel one of the users request for the data.



Programming and Software

SQL

SQL stands for Structured Query Language and is the standard programming language used to develop and access relational databases. It was originally developed to be an English-like query language which could be used by untrained end users, but it is in fact not like English and never achieved any success as a tool for end users. SQL did become a favorite with programmers and is today the language of choice in the database world.

There are two lines of thought as to how to pronounce SQL. One is to spell out the letters as in S-Q-L. Another is to pronounce it like the word "Sequel." SQL was originally spelled SEQUEL by the IBM engineers who created it in the early 1970s. The word SEQUEL stood for *S*tructured *E*nglish *Que*ry *L*anguage.

SQL is not an "open" language although the majority of SQL dialects are 90% compatible. Each vendor has some unique features and processes which are reflected in their version of SQL.

Middleware

Middleware refers to Client/Server software that exists between the client and the server. SQL middleware is software designed to eliminate the differences between various SQL servers. The SQL middleware interface allows client programs to access data stored on any supported server. It locates the data being requested, translates and forwards the request, receives the requested data and sends it to the client applications.

Database Servers

A database server refers to the software on which the database is run. It is almost always in a client/server environment and uses the SQL Query Language. Some examples are:

- **IBM* DB2** - Runs on Mainframes (MVS) to midrange systems (AS/400). Usually the operating system is UNIX or OS/2.
- **Oracle* Server** - Runs on almost any platform and operating system. Oracle is the largest vendor of database and database application software. Oracle is also a major proponent of the Network Computer (NC).
- **Microsoft* SQL Server** - Mid-range database (as of 1997) which runs exclusively on Windows* NT. Originally designed and developed in conjunction with Sybase*, the two companies have now gone their separate directions.
- **Informix* 4GL and Informix New Era** - Relational Database Management Systems that run on most UNIX* platforms.
- **Sybase* SQL Server** - Sybase specializes in Client/Server applications. Products including Sybase SQL Server as well as the Powerbuilder Application Development software.

Some of the functions of a Database Server include:

- **Locking** - Prevents conflicts when multiple users attempt to access the same data at the same time.
- **Deadlock detection** - Prevents a stalemate situation in which database users are waiting for resources which are in use by another user. If a deadlock situation is detected, one or more users are blocked or terminated so that the needed resources are available to complete the necessary transactions.
- **Performance Optimization** - Selects the most efficient way an SQL request can be processed.
- **Backup and Recovery** - Restores the database in the event of system failure.

Database Terminology

Just a few more terms which are good to know when discussing database issues.

DBMS - A Database Management System is the hardware and software on which the database is maintained.

Join - Involves retrieving multiple tables from a database by using the "select" statement in SQL.

Cursor - A cursor is a pointer to the "current row" of a SQL result set. The cursor allows a procedural programming language such as COBOL or C to retrieve and process the data one row at a time.

CLI - Call Level Interface. Used to request database services via a visual programming language such as Visual Basic. A CLI is used to call special independent SQL routines rather than using SQL statements that might be embedded directly into the program.

Examples of CLIs are:

- SAG CLI- The SQL Access Group created a standard CLI for SQL in 1992. The SAG CLI is also known as X/Open CLI.
- ODBC - Microsoft's Open Database Connectivity is a standard CLI for accessing SQL databases in a Windows* NT environment.
- IDAPI - Integrated Database Application Programming Interface. It is an alternative to ODBC. Was developed by Borland*, IBM and Novell*.

Client/Server - An architecture in which the client, a personal computer or workstation, requests data and/or processing from a server, both of which are connected to a network.

Fat/Thin Server and Fat/Thin Client - A client is considered "fat" if it performs most of the application processing and "thin" if it does not. A client may also be considered fat or thin depending on its role in the process.

Normalization - Normalization is the process of eliminating redundant information in a database design.

OLTP - Online Transaction Processing. OLTP applications are the mission critical applications that handle the day to day transactions such as order processing, shipping, account payables, etc.

OLAP - Online Analytical Processing. Decision support software that allows the user to analyze information that has been summarized. OLAP tools are used to perform trend analysis.

WOLAP - Web based OLAP

Object Oriented Database - A database that holds abstract data types (objects) and is managed by a DBMS. These databases are suited for data with complex relationships that are difficult to model. It is also capable of handling multimedia data types including images, audio and video files.

Primary Keys - Column or combination of columns which uniquely identifies every row in a table.

Schema - Pronounced “skeema,” it is the definition for an entire database. Schemas are often designed with visual modeling tools which automatically create the SQL code necessary to define table structures.

Transaction Processing - A set of database updates which must be processed as a set. None of the updates will take place should any of the data fail to be updated.

Three Tier Client/Server - Three way interaction in a Client/Server environment in which the user interface is stored in the client, the business application code is stored on one or more servers and the data is stored on a database server.

Two Tier Client/Server - Two way interaction in a Client/Server environment in which the user interface is stored on the client, the data is stored on a server and the business application may be stored on either the client or a server.

Key Points to Remember

- A Relational Database is any database which can establish relationships between different tables.
- SQL is the query language used to create and maintain a Relational Database
- Constants and Nulls are two ways in which data integrity is maintained.
- Referential Integrity checks the integrity of data in separate tables. Example - Customer names and identification numbers.
- Concurrency is the ability to allow multiple users access to a database simultaneously.
- Locking prevents users from altering data.
- A Distributed Database is spread across multiple servers.
- Three methods of distributing data are downloading, replication, and fragmentation.